4

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/696,968	10/30/2003	Joseph G. Laura	IDF 2562 (4000-15900)	8504

28003 7590 03/28/2007
SPRINT
6391 SPRINT PARKWAY
KSOPHT0101-Z2100
OVERLAND PARK, KS 66251-2100

EXAMINER	
KISS, ERIC B	
ART UNIT	**PAPER NUMBER**
2192	

SHORTENED STATUTORY PERIOD OF RESPONSE	MAIL DATE	DELIVERY MODE
3 MONTHS	03/28/2007	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

If NO period for reply is specified above, the maximum statutory period will apply and will expire 6 MONTHS from the mailing date of this communication.

Office Action Summary	Application No.	Applicant(s)	
	10/696,968	LAURA, JOSEPH G.	
	Examiner	Art Unit	
	Eric B. Kiss	2192	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *30 October 2003*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-43* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-43* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☒ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *01 March 2004* is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)	4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)	5) ☐ Notice of Informal Patent Application
3) ☒ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date *20040209, 20050531*.	6) ☐ Other: _____.

DETAILED ACTION

1. Claims 1-43 have been examined.

Claim Objections

2. Claim 1 is objected to because of the following informalities: "the comprising" in line 1 should presumably read "the system comprising". Appropriate correction is required.

Claim Rejections - 35 USC § 101

3. 35 U.S.C. 101 reads as follows:

> Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

4. Claims 1-43 are rejected under 35 U.S.C. 101 because the claimed invention is directed to non-statutory subject matter.

Descriptive material can be characterized as either "functional descriptive material" or "nonfunctional descriptive material." In this context, "functional descriptive material" consists of data structures and computer programs which impart functionality when employed as a computer component. (The definition of "data structure" is "a physical or logical relationship among data elements, designed to support specific data manipulation functions." The New IEEE Standard Dictionary of Electrical and Electronics Terms 308 (5th ed. 1993).) "Nonfunctional descriptive material" includes but is not limited to music, literary works and a compilation or mere arrangement of data. Both types of "descriptive material" are nonstatutory when claimed as descriptive material *per se*. *In re Warmerdam*, 33 F.3d 1354, 1361, 31 USPQ2d 1754, 1760 (claim to a data structure per se held nonstatutory).

Data structures not claimed as embodied in computer-readable media are descriptive material *per se* and are not statutory because they are not capable of causing functional change in the computer. *See, e.g.*, *In re Warmerdam*, 33 F.3d 1354, 1361, 31 USPQ2d 1754, 1760 (claim to a data structure per se held nonstatutory). Such claimed data structures do not define any structural and functional interrelationships between the data structure and other claimed aspects of the invention which permit the data structure's functionality to be realized. In contrast, a claimed computer-readable medium encoded with a data structure defines structural and functional interrelationships between the data structure and the computer software and hardware components which permit the data structure's functionality to be realized, and is thus statutory.

Similarly, computer programs claimed as computer listings per se, *i.e.*, the descriptions or expressions of the programs, are not physical "things." They are neither computer components nor statutory processes, as they are not "acts" being performed. Such claimed computer programs do not define any structural and functional interrelationships between the computer program and other claimed elements of a computer which permit the computer program's functionality to be realized. In contrast, a claimed computer-readable medium encoded with a computer program is a computer element which defines structural and functional interrelationships between the computer program and the rest of the computer which permit the computer program's functionality to be realized, and is thus statutory. *See In re Lowry*, 32 F.3d 1579, 1583-84, 32 USPQ2d 1031, 1035.

Claims 1-15 comprise a series of elements that can be reasonably interpreted as software, *per se*. The claim does not define any structural and functional interrelationships between the software elements and a computer that would permit the described functionality to be realized when the software is employed as a computer component. Accordingly, claims 1-15 appear to merely set forth functional descriptive material *per se*, which is nonstatutory.

A claim that requires one or more acts to be performed defines a process. However, not all processes are statutory under 35 U.S.C. § 101. To be statutory, a claimed process must either: (A) result in a physical transformation for which a practical application is either disclosed in the specification or would have been known to a skilled artisan, or (B) be limited to a practical application which produces a useful, tangible, and concrete result. *See Diamond v. Diehr*, 450 U.S. 175, 183-84, 209 USPQ 1, 9 (1981) (quoting *Cochrane v. Deener*, 94 U.S. 780, 787-88 (1876)) ("A [statutory] process is a mode of treatment of certain materials to produce a given result. It is an act, or a series of acts, performed upon the subject-matter to be transformed and reduced to a different state or thing . . . . The process requires that certain things should be done with certain substances, and in a certain order; but the tools to be used in doing this may be of secondary consequence."). *See also In re Alappat*, 33 F.3d 1526, 1543, 31 USPQ2d 1545, 1556-57 (quoting *Diehr*, 450 U.S. at 192, [209 USPQ at 10]).

In *State Street*, the Federal Circuit examined some of its prior section 101 cases, observing that the claimed inventions in those cases were each for a "practical application of an abstract idea" because the elements of the invention operated to produce a "useful, concrete and tangible result." *State St. Bank & Trust v. Signature Fin. Group*, 149 F.3d 1368, 1373-74, 47 USPQ2d 1596, 1601-02 (Fed Cir. 1998). For example, the court in *State Street* noted that the claimed invention in *Alappat* "constituted a practical application of an abstract idea (a mathematical algorithm, formula, or calculation), because it produced 'a useful, concrete and tangible result'—the smooth waveform." *Id.* Similarly, the claimed invention in *Arrhythmia* "constituted a practical application of an abstract idea (a mathematical algorithm, formula, or calculation), because it corresponded to a useful, concrete and tangible thing—the condition of a patient's heart." *Id.* (citing *Arrhythmia Research Tech. V. Corazonix Corp.*, 958 F.2d 1053, 22 USPQ2d 1033 (Fed. Cir. 1992)).

In determining whether the claim is for a "practical application," the focus is not on whether the steps taken to achieve a particular result are useful, tangible and concrete, but rather that the final result is "useful, tangible and concrete." The Federal Circuit further ruled that it is of little relevance whether a claim is directed to a machine or process for the purpose of a § 101 analysis. *AT&T Corp. v. Excel Commc'ns*, 172 F.3d 1352, 1358, 50 USPQ2d 1447, 1451 (Fed. Cir. 1999).

Claims 1-43 are directed to systems (claims 1-15) and methods (claims 16-43) for resolving memory space to an operable portion of a COBOL program (claims 1-22) and receiving, by part of a COBOL program, an address of a shared memory block (claims 23-43). This claimed subject matter lacks a practical application of a judicial exception (law of nature, abstract idea, naturally occurring article/ phenomenon) since it fails to produce a useful, concrete and tangible result. Specifically, the claimed subject matter does not produce a tangible result because the claimed subject matter fails to produce a result that is limited to having real world value rather than a result that may be interpreted to be abstract in nature as, for example, a thought, a computation, or manipulated data. More specifically, the claimed subject matter describes at best the performing of a process that is not tied to any particular tangible output capable of being, for example, stored, displayed, or conveyed in any manner causing any useful functional or structural change in a computer system so as to achieve a practical application.

This produced result remains in the abstract and, thus, fails to achieve the required status of having real world value.

5. To expedite a complete examination of the instant application, the claims rejected under 35 U.S.C. §101 (non-statutory) above are further rejected as set forth below in anticipation of Applicant amending these claims to place them within the four statutory categories of invention.

Claim Rejections - 35 USC § 112

6. The following is a quotation of the second paragraph of 35 U.S.C. 112:

> The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

7. Claims 24-26 are rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.

Claim 24 recites the limitation "The method of Claim 1" in line 1. There is insufficient antecedent basis for this limitation in the claim. In the interest of compact prosecution, the examiner subsequently interprets claim 24 as reading, "The method of Claim 23," based on the proximity of independent claim 23 to claim 24 (*See* MPEP § 608.01(n)(IV)).

Claim Rejections - 35 USC § 102

8. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (a) the invention was known or used by others in this country, or patented or described in a printed publication in this or a foreign country, before the invention thereof by the applicant for a patent.

9. Claims 1-43 are rejected under 35 U.S.C. 102(a) as being anticipated by "BEA Tuxedo®: Using the ATMI /Q Component," January 2003, BEA Systems, Inc. (hereinafter "[BEA2003]").

Regarding claim 1, [BEA2003] discloses: a system to enable queues for COBOL programs (see, e.g., p. 4-1), the system comprising:

a memory space (see, e.g., p. 1-1 (disk or memory));

an operating system maintaining a key related to an address of the memory space (see, e.g., "Queue Space Names, Queue Names, and Service Names," beginning on p. 2-4);

a COBOL routine maintaining the key in an index, the COBOL routine communicating with the operating system to receive the address of the memory space based on the key (see, e.g., "Queue Space Names, Queue Names, and Service Names," beginning on p. 2-4);

a COBOL program coupled to receive the memory address based on the key from the index of the COBOL routine (see, e.g., "Emphasis on the Default Case," beginning on p. 4-2).

Regarding claim 2, [BEA2003] further discloses the COBOL program operably calls the COBOL routine using an identifier and wherein the index maintains the identifier associated with the key (see, e.g., "Queue Space Names, Queue Names, and Service Names," beginning on p. 2-4).

Regarding claim 3, [BEA2003] further discloses the index maintains a plurality of identifiers each associated with one of a plurality of keys maintained by the index (see, e.g., "Queue Space Names, Queue Names, and Service Names," beginning on p. 2-4).

Regarding claim 4, [BEA2003] further discloses the plurality of identifiers are further defined as an alphanumeric identifier (see, e.g., "Queue Space Names, Queue Names, and Service Names," beginning on p. 2-4).

Regarding claim 5, [BEA2003] further discloses the plurality of identifiers are further defined as names (see, e.g., "Queue Space Names, Queue Names, and Service Names," beginning on p. 2-4).

Regarding claim 6, [BEA2003] further discloses the COBOL program receives the memory address via a linkage section of the COBOL program (see, e.g., "Emphasis on the Default Case," beginning on p. 4-2).

Regarding claim 7, [BEA2003] further discloses the COBOL program is operable to receive the memory address to enable the COBOL program to resolve the information in the memory space to the linkage section of the COBOL program (see, e.g., "Emphasis on the Default Case," beginning on p. 4-2).

Regarding claim 8, [BEA2003] further discloses the memory space is operable for a message queue (see, e.g., pp. 1-2 and 1-3).

Regarding claim 9, [BEA2003] further discloses a coordination module operable to receive a request from the COBOL program to read and write information to the message queue (see, e.g, "Enqueueing Messages," beginning on p. 4-3; "DequeueingMessages," beginning on p. 4-17).

Regarding claim 10, [BEA2003] further discloses the coordination module coordinates reading and writing information to the message queue in a last-in-first-out order (see, e.g., p. 1-4 (LIFO)).

Regarding claim 11, [BEA2003] further discloses the coordination module coordinates reading and writing information to the message queue in a first-in-first-out order (see, e.g., p. 1-4 (FIFO)).

Regarding claim 12, [BEA2003] further discloses the memory space is operable for a memory queue (see, e.g., p. 1-4).

Regarding claim 13, [BEA2003] further discloses a coordination module is operable to prevent a conflict (see, e.g, "Error Handling", beginning on p. 1-9).

Regarding claim 14, [BEA2003] further discloses the coordination module is operable from a call to an operating system (see, e.g., pp. 1-2 and 1-3).

Regarding claim 15, [BEA2003] further discloses the coordination module is operable to prevent writing when the memory space is full and further operable to prevent reading when the memory space is empty (see, e.g., "Using Queue Capacity Limits", beginning on p. 2-13).

Regarding claim 16, [BEA2003] discloses a method of enabling queues for COBOL programs, comprising:

creating a queue using a memory space (see, e.g., p. 1-1 (disk or memory));

providing an operating system having a key related to an address of the memory space; maintaining the key in an index (see, e.g., "Queue Space Names, Queue Names, and Service Names," beginning on p. 2-4);

communicating with the operating system to receive the address of the memory space based on the key (see, e.g., "Queue Space Names, Queue Names, and Service Names," beginning on p. 2-4);

resolving the memory space to an operable portion of the COBOL program based on the key from the index (see, e.g., "Emphasis on the Default Case," beginning on p. 4-2).

Regarding claim 17, [BEA2003] further discloses a COBOL routine maintains the index and receives the address of the memory space from the operating system (see, e.g., "Queue Space Names, Queue Names, and Service Names," beginning on p. 2-4).

Regarding claim 18, [BEA2003] further discloses the COBOL routine is further defined as a COBOL technical layer having a plurality of routines, the method further comprising:

attaching to an existing queue (see, e.g., "Using Queue Capacity Limits", beginning on p. 2-13);

querying the queue to determine whether the queue exists and to determine the size of the queue (see, e.g., "Using Queue Capacity Limits", beginning on p. 2-13);

adding, by a push module of the COBOL technical layer, at least one row to the queue (see, e.g., "Enqueueing Messages," beginning on p. 4-3);

blocking when the queue is full (see, e.g., "Using Queue Capacity Limits", beginning on p. 2-13);

removing, by a pop module of the COBOL technical layer, a top row from the queue (see, e.g., "Dequeueing Messages," beginning on p. 4-17);

detaching from a queue (see, e.g., "Using Queue Capacity Limits", beginning on p. 2-13); and

removing a queue from a system (see, e.g., "Using Queue Capacity Limits", beginning on p. 2-13).

Regarding claim 19, [BEA2003] further discloses the COBOL technical layer is further defined as a COBOL library wherein the routines are callable from the COBOL program (see, e.g., "Introduction," beginning on p. 4-1).

Regarding claim 20, [BEA2003] further discloses the COBOL technical layer is integral to the COBOL program (see, e.g., "Introduction," beginning on p. 4-1).

Regarding claim 21, [BEA2003] further discloses the COBOL technical layer is further defined as enabled by a COBOL compiler (see, e.g., "Introduction," beginning on p. 4-1 (the functions are part of the ATMI COBOL language and its associated compiler).

Regarding claim 22, [BEA2003] further discloses the compiler enabled functionality is further defined as pre-compiler enabled functionality (see, e.g., "Introduction," beginning on p. 4-1 (the functions are part of the ATMI COBOL language)).

Regarding claim 23, [BEA2003] discloses a method of sharing memory between COBOL programs, comprising:

requesting, by a first and second COBOL programs, an address of a block of memory (see, e.g., "Queue Space Names, Queue Names, and Service Names," beginning on p. 2-4; p. 1-5 (peer-to-peer));

returning the address to the first and second COBOL programs (see, e.g., "Queue Space Names, Queue Names, and Service Names," beginning on p. 2-4; "Emphasis on the Default Case," beginning on p. 4-2); and

sharing, by the first and second COBOL programs, the block of memory (see, e.g., p. 1-5 (describing sharing memory)).

Regarding claim 24, [BEA2003] further discloses allocating the block of shared memory (see, e.g., "Creating a Queue Space: qspacecreate," beginning on p. 2-10).

Regarding claim 25, [BEA2003] further discloses an operating system allocates the block of shared memory based on parameters obtained from the first and second COBOL programs at compilation (see, e.g., "Dynamic Configuration" on p. 2-8).

Regarding claim 26, [BEA2003] further discloses the operating system maintains the address of the block of shared memory and provides the address in response to the first and second COBOL programs requesting the address (see, e.g., "Queue Space Names, Queue Names, and Service Names," beginning on p. 2-4).

Regarding claim 27, [BEA2003] further discloses the first and second COBOL programs request the address of the block of memory using an identifier (see, e.g., "Queue Space Names, Queue Names, and Service Names," beginning on p. 2-4).

Regarding claim 28, [BEA2003] further discloses an index maintaining the identifier associated with a key, the key used to obtain the address of the block of memory from an operating system (see, e.g., "Queue Space Names, Queue Names, and Service Names," beginning on p. 2-4; "Emphasis on the Default Case," beginning on p. 4-2).

Regarding claim 29, [BEA2003] further discloses the first and second COBOL programs use the address to map a linkage section of each of the first and second COBOL programs to the block of memory to enable the first and second COBOL programs to share the block of memory (see, e.g., "Queue Space Names, Queue Names, and Service Names," beginning on p. 2-4; "Emphasis on the Default Case," beginning on p. 4-2).

Regarding claim 30, [BEA2003] discloses a method of sharing memory between COBOL programs, the method comprising:

maintaining, by a COBOL routine, an index of shared memory addresses (see, e.g., "Queue Space Names, Queue Names, and Service Names," beginning on p. 2-4);

requesting, by a COBOL program, a shared memory block (see, e.g., "Enqueueing Messages," beginning on p. 4-3); and

receiving to a linkage section of the COBOL program an address of the shared memory block from the COBOL routine (see, e.g., "Enqueueing Messages," beginning on p. 4-3 "Queue Space Names, Queue Names, and Service Names," beginning on p. 2-4; "Emphasis on the Default Case," beginning on p. 4-2).

Regarding claim 31, [BEA2003] further discloses the COBOL routine is a part of at least the COBOL program (see, e.g., "Introduction," beginning on p. 4-1 (the functions are part of the ATMI COBOL language)).

Regarding claim 32, [BEA2003] further discloses the COBOL routine further comprises a plurality of COBOL subroutines (see, e.g., "Introduction," beginning on p. 4-1 (the functions are part of the ATMI COBOL language)).

Regarding claim 33, [BEA2003] further discloses the COBOL routine is a library routine callable from the COBOL program (see, e.g., "Introduction," beginning on p. 4-1 (the functions are part of the ATMI COBOL language)).

Regarding claim 34, [BEA2003] further discloses the COBOL routine is a COBOL function enabled by a COBOL compiler (see, e.g., "Introduction," beginning on p. 4-1 (the functions are part of the ATMI COBOL language and its associated compiler)).

Regarding claim 35, [BEA2003] further discloses creating, by the COBOL routine, a shared memory block (see, e.g., "Enqueueing Messages," beginning on p. 4-3).

Regarding claim 36, [BEA2003] further discloses creating the shared memory block further comprises calling the operating system to from the COBOL routine to request a block of memory (see, e.g., pp. 1-2 and 1-3).

Regarding claim 37, [BEA2003] further discloses attaching the COBOL program to the shared memory block (see, e.g., "Emphasis on the Default Case," beginning on p. 4-2).

Regarding claim 38, [BEA2003] further discloses maintaining, by the COBOL routine, an index having an identifier associated with the address of the shared memory block (see, e.g., "Queue Space Names, Queue Names, and Service Names," beginning on p. 2-4).

Regarding claim 39, [BEA2003] further discloses searching the index based on the identifier to locate the address of the shared memory block associated with the identifier (see, e.g., "Queue Space Names, Queue Names, and Service Names," beginning on p. 2-4).

Regarding claim 40, [BEA2003] further discloses the searching is accomplished by the COBOL routine in response to receiving a request from the COBOL program, the request including the identifier (see, e.g., "Queue Space Names, Queue Names, and Service Names," beginning on p. 2-4).

Regarding claim 41, [BEA2003] further discloses the searching is accomplished by the COBOL program (see, e.g., "Queue Space Names, Queue Names, and Service Names," beginning on p. 2-4; "Emphasis on the Default Case," beginning on p. 4-2).

Regarding claim 42, [BEA2003] further discloses the shared memory is defined as protected, and the method further comprises calling a semaphore routine to manage modifications to the shared memory (see, e.g., "Creating a Queue Space: qspacecreate," beginning on p. 2-10).

Regarding claim 43, [BEA2003] further discloses creating the shared memory (see, e.g., "Creating Queue Spaces and Queues," beginning on p. 2-8);

attaching to the shared memory (see, e.g., "Using Queue Capacity Limits", beginning on p. 2-13);

detaching from the shared memory (see, e.g., "Using Queue Capacity Limits", beginning on p. 2-13);

removing the shared memory (see, e.g., "Using Queue Capacity Limits", beginning on p. 2-13); and

querying the shared memory to determine whether the shared memory exists and a size of the shared memory (see, e.g., "Using Queue Capacity Limits", beginning on p. 2-13).

Conclusion

10. Any inquiry concerning this communication or earlier communications from the Examiner should be directed to Eric B. Kiss whose telephone number is (571) 272-3699. The Examiner can normally be reached on Tue. - Fri., 7:00 am - 4:30 pm. The Examiner can also be reached on alternate Mondays.

If attempts to reach the Examiner by telephone are unsuccessful, the Examiner's supervisor, Tuan Dam, can be reached on (571) 272-3695. The fax phone number for the organization where this application or proceeding is assigned is (571) 273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Any inquiry of a general nature should be directed to the TC 2100 Group receptionist: 571-272-2100.

[illegible]

Eric B. Kiss
March 24, 2007